33

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/928,881	08/13/2001	Konstantin Konson	DE920000074US1	8141

48115 7590 05/20/2005

HOLLAND & KNIGHT LLP
701 BRICKELL AVE
SUITE 3000
MIAMI, FL 33131

EXAMINER	
TSAI, SHENG JEN	

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 05/20/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/928,881	Applicant(s) KONSON ET AL.	
	Examiner Sheng-Jen Tsai	Art Unit 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 April 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Tle

DETAILED ACTION

1. This Office Action is taken in response to Applicants' Amendment and Remarks filed on April 1, 2005 regarding application 09/928,881 filed on August 13, 2001.

2. Claims 1-24 are pending in the application under consideration.

Claims 1-24 have been amended.

3. ***Response to Amendments and Remarks***

Applicants' amendments and remarks have been fully and carefully considered with examiner's responses detailed below.

As to amendment and remark for claims 1:

Applicants contend that the fourth element of claim 1 includes the term "an object" and serves as the antecedent for "said object description." The examiner concurs with this observation. Therefore the 35 U.S.C 112 rejection is withdrawn.

As to amendment and remark for claims 14:

Applicants contend that the segment map and object map cited by the examiner using the prior art (Reiter et al., US 5,752,243) is not appropriate because the construction of Reiter et al.'s segment map is a tree-like structure while the segment map recited in Applicants' invention is not built as a tree. The examiner disagrees with this assessment.

It should be noted that claim analysis is performed based on the language and wording recited in each claim. Since claim 1 does not specify the structure of either the segment map or the object map, any kind of segment map and object map, regardless their structure, would be considered as meeting the requirements of the claim as long

as their characteristics and properties read on the description of the claim language, as the examiner pointed out in the "As to claim 14" section of the claim analysis in the previous Office action.

Applicants also contend that their invention does not concern the space availability issue and rather specifies method and steps to store the object. Again, the language of claim 14 does not recite "space availability" at all. Therefore, whether the space availability is an issue or not regarding the prior art is of no consequence as far as claim analysis is concerned.

Applicants also contend that Reiter et al.'s data structure shown in figure 5 does not qualify as the "object description" as recited by claim 14, which reads "creating an object description for an object by saving values owned by the object of the variables belonging to its class." However, figure 5 of Reiter et al. indeed shows that the data structure stores the values of a plurality of variables belonging to the object; hence qualifies as object descriptor.

Applicants further contend that Reiter et al. fail to teach adding a new element as recited in claim 14. However Reiter et al. indeed teach this aspect [to insert a unit of data into the MDB-tree, ... (column 9, lines 1-25)].

Therefore, the examiner's position regarding this claim, and all those claims dependent from it, remains the same as stated in the previous Office action.

Claim Objections

4. Claim 1 is objected to because of the following informalities:

The amended claim 1 recites, first, "retrieving from said persistent storage a second list comprising second reference to blocks, whereby said **blocks** contain an **object description**," and then "creating an object in said volatile memory using said object description from said **segment** and saving ..."

The examiner notes that Applicants make a distinction between a segment and a block in claim 1 as well as the rest of the claims, and believes that Applicants do not intend to treat them as the same. Therefore, it creates confusion and ambiguity when one part of the claim states that "object description is contained in a block" and, at the same time, another portion of the claim states that "object description from a segment." Appropriate correction is required to clarify the issue.

In the subsequent claim analysis, the examiner will interpret that "object description" is contained in a block instead of a segment and treats the claim language as "creating an object in said volatile memory using said object description from said **block** and saving ..."

5. Claim Rejections - 35 USC § 102

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1- are rejected under 35 U.S.C. 102(b) as being anticipated by Reiter et al. (U.S. 5,752,243).

As to claim 1, Reiter et al. disclose **a method for restoring persistently stored objects** [a computer method and storage structure for storing and accessing multidimensional data is provided (abstract); figure 2 shows the persistent secondary

storage unit (item 23)] **of an object-oriented environment** [figure 5 shows an instance of the object oriented environment] **established in a computer system** [a computer method and storage structure for storing and accessing multidimensional data is provided (abstract); figure 2 shows the CPU unit (item 25)] **having a volatile memory** [figure 2 shows the volatile memory unit, the RAM (item 22)] **and a persistent storage** [figure 2 shows the persistent secondary storage unit (item 23)], **the method comprising the steps of**:

retrieving from said persistent storage [an MDB-tree is read from the secondary storage device (figure 2, 23) in page units (column 6, lines 56-64)] **a first list comprising first references to segments, stored in said persistent storage** [figure 4 shows how the memory are allocated, including the first list (key value table) containing references to the segment (figure 3A, item 26; figure 3D); the nodes are indexed by a primary key value (the first list) while the subnodes in a subtree are indexed by a secondary key values (the second list) (abstract)];

retrieving all segments referenced by said first references and storing them in said volatile memory [figure 4 shows how the memory are allocated, including the data area for storing the segments (figure 3A, item 28); a tree manager provided by the present invention stores data such as pointers, variable length data records, other B-trees, and directories in a multidimensional B-tree (column 3, lines 3-6)];

saving in said first list the difference between the an old memory address at which the segment used to reside in the volatile memory, and the a new memory address at which said segment is stored;

retrieving from said persistent storage a second list comprising second references to blocks, whereby said blocks contain an object description [figure 4 shows how the memory are allocated, including the second list (subnode table) containing references to blocks (figure 3A, item 27; figure 3D); the nodes are indexed by a primary key value (the first list) while the subnodes in a subtree are indexed by a secondary key values (the second list) (abstract); figure 4 shows how the memory are allocated, including the data area associated with a block (subnode) of a segment (Key value table) as shown in figure 3A, item 28; because secondary storage is often times logically divided into fixed-size blocks or pages, units of data in an MDB-tree are physically divided into page-size subtrees (column 3, lines 10-12); an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64); figure 5 shows the object description containing variables are associated with each object and block (for examples, item 118, 120, 122, 124, 126, 138, 144 and 146)];

determining which segment contains said block referenced by a particular element of said second list [figure 5 shows the use of pointers for one object to reference another object and the associated object description, which implies that the address of the object and the object description being referenced has to be determined so that the pointer can be set accordingly];

creating an object in said volatile memory using said object description from said segment and saving a new address of said created object in said second list in volatile memory [figure 8B, a continuation of the process of adding a new element as

shown in figure 8A, shows the step of storing pointer to the new page/block in pointer node, hence setting the address of the object description; the entire storage structure shown in figure 4, including the subnode table (the second list), is stored back to the secondary storage after the operations of insertion, modification, and/or deletion are performed at RAM; an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64)]; **initializing said new object with values taken from said object description** [figure 8B, step 172, select subtree; step 174 create pointer node on current page; figure 5]; and determining the said new addresses of said new object referenced by the newly created object [figure 8B, step 176 shows storing pointer (i.e., new address) to pointer node (i.e., the new object) in subnode table of selected parent node]; **and setting said new address as the reference in said new object** [figure 8B, step 176 shows storing pointer (i.e., new address) to pointer node (i.e., the new object) in subnode table of selected parent node].

As to claims 2 and 3, Reiter et al. teach that **the first** (the key value table shown in figure 4) **and/or the second list** (the subnode table shown in figure 4) are organized using a multidimensional **B-tree structure** (abstract), as shown in figures 3A, 3B, 3C, and 3D. Note that B-tree is a special type of **ordered lists**.

As to claim 4, Reiter et al. teach that **the elements of the first ordered list are indexed by the first reference** [the nodes are indexed by a primary key value while the subnodes in a subtree are indexed by secondary key values (abstract)].

As to claims 5 and 7, Reiter et al. teach that **each of the first/second references corresponds to the old memory address at which the respective segment/block used to reside in the volatile memory** [column 10, lines 50-67].

As to claim 6, Reiter et al. teach that **the elements of the second ordered list are indexed by the second reference** [the nodes are indexed by a primary key value while the subnodes in a subtree are indexed by secondary key values (abstract)].

As to claim 8, Reiter et al. teach that **object description is formed by a collection of values owned by an object for the variables belonging to its class** [figures 4, 5, 6, and 7].

As to claim 9, Reiter et al. teach that **each value in said object description of variables having a variable length the method further comprising the steps of:**
allocating a number of blocks that allows to keep the actual value of the variable having variable length;
creating a linked list of said number of blocks;
saving said value into said number of blocks; and
storing a reference to the head of the linked list in said object description [figure 5].

As to claims 10-12, Reiter et al. teach that searching for a particular key value in a B-tree is similar to searching a binary tree, except that instead of making a two-way branching decision at each non-leaf node (i.e., deciding which of the two child nodes to examine), a multi-way branching decision is made. The multi-way branching decision depends upon the node's child nodes. As stated above, for a node x containing n[x]

key values, node x has n [x]+1 child nodes. Therefore, at each node x, an (n[x]+1)-way branching decision is made (column 2, lines 16-25); figure 5].

As to claim 13, Reiter et al. teach that **all reference to heads of linked lists comprising the steps of:**

Reading all blocks of said linked list;

Allocating memory to store the value of variables retrieved from the linked list; and

Storing the value in the allocated memory [figure 5; column 1, lines 32-43; column 2, lines 54-67; column 3, lines 1-20].

As to claim 14, Reiter et al. disclose **a method for persistently storing objects** [a computer method and storage structure for storing and accessing multidimensional data is provided (abstract); figure 2 shows the persistent secondary storage unit (item 23)] **of an object oriented environment** [figure 5 shows an instance of the object oriented environment] **established on a computer system** [a computer method and storage structure for storing and accessing multidimensional data is provided (abstract); figure 2 shows the CPU unit (item 25)] **having a volatile memory** [figure 2 shows the volatile memory unit, the RAM (item 22)] **and a persistent storage** [figure 2 shows the persistent secondary storage unit (item 23)], **the method comprising the steps of:**

allocating in said volatile memory segments, that is, pieces of memory [figure 4 shows how the memory are allocated, including the data area for storing the segments (figure 3A, item 28); a tree manager provided by the present invention stores data such

as pointers, variable length data records, other B-trees, and directories in a multidimensional B-tree (column 3, lines 3-6)];

creating a first list (segment map) containing first references to said segments [figure 4 shows how the memory are allocated, including the first list (key value table) containing references to the segment (figure 3A, item 26; figure 3D); the nodes are indexed by a primary key value (the first list) while the subnodes in a subtree are indexed by a secondary key values (the second list) (abstract)];

creating a second list (object map) containing second references to blocks, that is, portions of said segments [figure 4 shows how the memory are allocated, including the second list (subnode table) containing references to blocks (figure 3A, item 27; figure 3D); the nodes are indexed by a primary key value (the first list) while the subnodes in a subtree are indexed by a secondary key values (the second list) (abstract)];

allocating a block of one of said segments [figure 4 shows how the memory are allocated, including the data area associated with a block (subnode) of a segment (Key value table) as shown in figure 3A, item 28; because secondary storage is often times logically divided into fixed-size blocks or pages, units of data in an MDB-tree are physically divided into page-size subtrees (column 3, lines 10-12); an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64)],

creating an object description by saving the values owned by the object of the variables belonging to its class into said allocated block [figure 5 shows the object

description containing variables are associated with each object and block (for examples, item 118, 120, 122, 124, 126, 138, 144 and 146)];

adding a new element to said second list containing the particular reference to said created object description [figure 8A shows the flow diagram of the process of adding a new element, item 160 indicates the step of selecting page (block) in which the new element unit should be added];

determining the address of the object description of another object referenced in said object [figure 5 shows the use of pointers for one object to reference another object and the associated object description, which implies that the address of the object and the object description being referenced has to be determined so that the pointer can be set accordingly] ;

setting the address of said respective object description as the reference in the created object description [figure 8B, a continuation of the process of adding a new element as shown in figure 8A, shows the step of storing pointer to the new page/block in pointer node, hence setting the address of the object description];

storing said second list (object map) on said persistent storage [the entire storage structure shown in figure 4, including the subnode table (the second list), is stored back to the secondary storage after the operations of insertion, modification, and/or deletion are performed at RAM; an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64)] ;

storing the segments referenced by said first list (segment map) on said persistent storage [the entire storage structure shown in figure 4, including the

segment (the data area), is stored back to the secondary storage after the operations of insertion, modification, and/or deletion are performed at RAM; an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64)]; **and**

storing said first list (segment map) on said persistent storage [the entire storage structure shown in figure 4, including the key value table (the first list), is stored back to the secondary storage after the operations of insertion, modification, and/or deletion are performed at RAM; an MDB-tree is written to the secondary storage device and read from the secondary storage device in page/block units (column 6, lines 62-64)].

As to claims 15 and 16, Reiter et al. teach that **the first** (the key value table shown in figure 4) **and/or the second list** (the subnode table shown in figure 4) are organized using a multidimensional **B-tree structure** (abstract), as shown in figures 3A, 3B, 3C, and 3D. Note that B-tree is a special type of **ordered lists**.

As to claim 17, refer to "As to claim 4."

As to claims 18 and 20, refer to "As to claims 5 and 7."

As to claim 19, refer to "As to claim 6."

As to claims 21-22, refer to "As to claims 10-12."

As to claim 23, refer to "As to claim 13."

As to claim 24, refer to "As to claim 1" and "As to claim 14."

Conclusion

7. Claims 1-24 are rejected as explained above.

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

May 16, 2005

PIERRE BATAILLE
PRIMARY EXAMINER
5/17/05

PIERRE BATAILLE
PRIMARY EXAMINER